JOURNAL OF COMPUTER AND SYSTEM SCIENCES 5, 511–523 (1971)

# Marked Directed Graphs*

F. COMMONER AND A.W. HOLT

*Applied Data Research, Wakefield, Massachusetts 01880*

AND

S. EVEN, A. PNUELI

*The Weizmann Institute of Science, Rehovot, Israel*

Received October 6, 1970

## I. INTRODUCTION

Diverse graph structure models for concurrent processing systems have been suggested and used. The structures differ in generality and scope according to the properties one wishes to model and analyze. In this paper we solve a problem of maximal storage requirements for a simple flowchart model called the Marked Graph Model.

Marked Graphs can be derived as a special case of a more general structure called a Petri Net. Petri Nets, a more powerful model, have been introduced and studied by C. A. Petri [1] and A. W. Holt [2]. A recent application of the Petri Net model is that of R. M. Shapiro and H. Saint [3]. Marked Graphs as restricted Petri Nets have been studied by H. Genrich [7] and Holt and Commoner [6]. Since no published account of their work is available yet, we include here some of the basic theory necessary for the understanding of our own contributions.

Marked Graphs are also introduced in a more generalized form by Karp and Miller [8]. They associate with each arc $d_p$ an input quantum $U_p$ and output quantum $W_p$ which are arbitrary nonnegative integers. In our model $U_p = W_p = 1$ for each arc. This simplification makes our model more amenable to analysis, and permits algorithmic answers to problems which in their model were quite complex to solve, e.g., the termination problem.

Similar restrictions are made by R. Reiter in his work [9] on scheduling and sequencing parallel processes. Even though the marked graph model is restricted, mainly in

* This research was partly supported by Advanced Research Projects Agency of the Department of Defense and was monitored by the U. S. Army Research Office-Durham N.C. 27706, under contract DAHC04 68 C 0043.

being able to model only those processes in which computation flow is data-independent and therefore a priori determinate, both Reiter's work on scheduling and ours on space requirements demonstrate that it can model interesting features of concurrent systems.

In Section II we describe live and safe marking, in Section III we discuss relations between markings, and in Section IV we describe an algorithm for finding a maximum marking of a given family. The results of Section II are those of Genrich (private communication, 1968). Most of the results of Section III were developed both by Genrich [7] and Holt and Commoner [5, 6], but we have described them in our own style and produced original and more algorithmic proofs. Section IV is original. We want to thank Professor G. B. Dantzig and Mr. I. Adler, of Stanford University, for their patience and advice.

## II. Live and Safe Markings of a Directed Graph

Assume we have a finite directed graph $G(V, E)$, where $V$ is the set of vertices and $E$ is the set of edges. The notation $a \xrightarrow{e} b$ means that the edge $e$ comes out of vertex $a$ and enters vertex $b$. We assign a number $M(e)$ of *tokens* (a nonnegative integer) to each edge $e$. The function $M$ is called a *marking* of the graph. A vertex is said to be *fireable* if the number of tokens on every one of its incoming edges is positive. The *firing* of a fireable vertex consists of taking off one token from each incoming edge, and adding one token to each outgoing edge. Since the number of incoming and outgoing edges is not necessarily the same, the total number of tokens on the graph may increase or decrease through firing.

We may consider the number of tokens on a simple directed circuit $C$; this number, $\langle M \mid C \rangle$, is the sum of tokens on the edges of the circuit. The following lemma is a direct consequence of the definition of firing.

LEMMA 1. *The token count of a directed circuit does not change by vertex firing.*

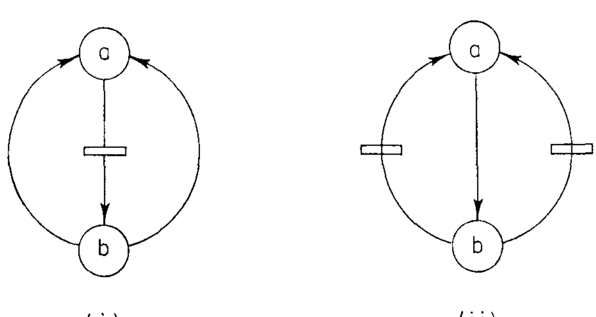


FIG. 1. Description of Example 1.

EXAMPLE 1. A marked graph is shown in Fig. 1(i). There is only one token in this marking, and it is on the edge from vertex $a$ to vertex $b$. Vertex $b$ is fireable. After firing vertex $b$, the two edges from $b$ to $a$ have one token on each of them and $a$ is fireable. After firing $a$ again we return to the original marking. Thus, the number of tokens in the graph changes from 1 to 2 and back to 1; but the token count on each of the two simple circuits remains 1.

A marking is called *live* if every vertex is fireable, or can be made fireable through some sequence of firings.

THEOREM 1. *A marking is live if and only if the token count of every directed circuit is positive.*

*Proof.* If the token count of some directed circuit is zero, no vertex on this circuit is fireable; since the token count does not change if other vertices are fired (Lemma 1), no vertex on this circuit can be made fireable through firings.

Now assume that the token count of every directed circuit is positive. Let $v$ be any vertex of the graph. Consider the token-free edges entering $v$. If there are none the vertex is fireable. If not, consider the vertices from which these edges emanate. If each of those is immediatelhy fireable, then, clearly, $v$ will become firebale after every one of them is fired. If some are not, consider the token-free edges entering them, etc. As we continue this backtracking, we are selecting a subgraph of $G$ which consists of $v$, the token-free edges entering $v$, the vertices from which these edges emanate, the token-free edges entering them, etc. The process must terminate, since $G$ is finite. Now, this subgraph must be circuit-free since there are no token-free directed circuits. Thus, the subgraph must have at least one vertex which has no incoming edges which belong to the subgraph. This vertex is fireable in the present marking of $G$. After firing it, the subgraph of the token-free-backtracking from $v$ is reduced by one vertex. By repeating this process, we can make $v$ fireable. Q.E.D.

COROLLARY 1. *A marking which is live remains live after firing.*

*Proof.* Since the token count of circuits does not change with firing (Lemma 1), and if a marking is live, then the count for all directed circuits is positive (Theorem 1), this count will stay positive after firing. By Theorem 1 the marking stays live. Q.E.D.

A marking is called *safe* if no edge is assigned more than one token, and if no sequence of firings can bring two tokens or more to one edge.

THEOREM 2. *A live marking is safe if and only if every edge in the graph is in a directed circuit with token count* 1.

*Proof.* If for every edge there exists a circuit, in which this edge takes part, with exactly one token on it, then by Lemma 1 the token count of this circuit remains 1, and therefore there will never be two or more tokens placed on this edge.

Assume that there exists an edge $e$, $a \xrightarrow{e} b$, such that all directed circuits which go through it have a token count of 2 or more. We want to demonstrate that by a proper sequence of firings we can place two tokens on $e$. If there are no tokens on $e$, we backtrack the token-free subgraph, starting with vertex $a$, as in the proof of Theorem 1. Thus, as in the proof of Theorem 1, we can make vertex $a$ fireable and fire it. This places one token on $e$. We repeat this construction. Again, the token-free subgraph backtracked from $a$ does not include $b$, since this would imply the existence of a circuit of token count 1 through $e$. Thus, we can fire vertex $a$ again without firing $b$, and therefore place a second token on $e$. Therefore, the initial marking is not safe. Q.E.D.

COROLLARY 2. *If a graph has a live and safe marking, then for every edge of the graph one can find a circuit which goes through the edge.*

This is an immediate consequence of Theorem 2. It eliminates the possibility of having sinks or sources or any separating edges in a graph which can be assigned with a live and safe marking. In fact, a much stronger statement is true:

THEOREM 3. *If the underlying undirected graph*[1] *of a directed graph $G(V, E)$ is connected, and if $G$ can be assigned with a live and safe marking, then $G$ is strongly connected.*

*Proof.* Assume that the underlying undirected graph of $G$ is connected, and that $G$ can be assigned with a live and safe marking. If $G$ is not strongly connected then there are two vertices $a$ and $b$ such that there is no directed path from $a$ to $b$ in $G$. Let $A$ be the set of all vertices, including $a$, which are reachable from $a$. Since $A \neq V$ and since the underlying undirected graph is connected, there must exist a vertex $a'$ in $A$ and a vertex $b'$ in $V - A$ which are connected by an edge $e$. The edge $e$ must be oriented $b' \xrightarrow{e} a'$, or $b'$ would belong to $A$. However, Corollary 2 implies the existence of a directed path from $a'$ to $b'$, which again contradicts the fact that $b' \in V - A$. Thus, $G$ must be strongly connected. Q.E.D.

The summary of Theorems 1, 2 and 3 is that a given marking of a graph is live and safe if and only if no directed circuit is token-free, and through every edge there is a circuit of token count 1. Also, if a live and safe marking exists, the graph is strongly connected. A natural question arises. Does there exist a live and safe marking for every strongly connected graph. This problem was known for some time as "Holt's toll-booth problem", and was settled by Genrich in 1969. This is the subject of our next theorem.

[1] That is, the graph resulting from $G$ by ignoring directions.

THEOREM 4. *For every finite, directed, strongly connected graph there exists a live and safe marking.*

*Proof.* Clearly we can find a live marking simply by putting one token on each edge. By Theorem 1 this marking is live. Now we can use the technique developed in the proofs of Theorems 1 and 2 to change the marking until it becomes safe, without changing its liveness. Assume that for a given edge the least token count for the directed circuits through it is $k > 1$. We can describe a sequence of firings that will bring $k$ tokens to the edge. By lifting $k - 1$ of them no circuit becomes token-free, and there is now a circuit through this edge with a token count 1. This can be repeated as long as there are edges for which no circuit of token count 1 exists. Q.E.D.

## III. RELATIONS BETWEEN MARKINGS

Let $\tilde{\sigma} = (v_{i_1}, v_{i_2}, \ldots, v_{i_l})$ be a firing sequence. We denote by $\sigma_i$ the number of times that vertex $v_i$ is fired in $\tilde{\sigma}$; that is, the number of times it appears in the sequence. Let $\bar{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_n)$, which is the vector of the firing numbers $\sigma_i$, and where $n = |V|$ (the number of vertices). Using this notation, we get that if $\tilde{\sigma}$ is a legal firing sequence leading from a marking $M$ to a marking $M'$, then for each arc $v_i \xrightarrow{e} v_j$, $M'(e) = M(e) + \sigma_i - \sigma_j$. It is interesting that the converse is also true.

THEOREM 5. *If a live marking $M$ and a vector $\bar{\sigma}$, with nonnegative integral components $\sigma_i$ for every vertex $v_i$, satisfy the condition $M(e) + \sigma_i - \sigma_j \geqslant 0$ for every edge $v_i \xrightarrow{e} v_j$, then there exists a firing sequence, legal for $M$, whose vector of firing numbers is $\bar{\sigma}$.*

*Proof.* Let $M$ be any marking and $\bar{\sigma}$ be a nonnegative integral vector which satisfies $M(e) + \sigma_i - \sigma_j \geqslant 0$ for every $v_i \xrightarrow{e} v_j$. We have to show that the marking $M'$ defined by $M'(e) = M(e) + \sigma_i - \sigma_j$ is achievable through a firing sequence $\tilde{\sigma}$ such that $\bar{\sigma}$ is $\tilde{\sigma}$'s vector of firing numbers.

If $\bar{\sigma}$ is the zero vector, then the statement is trivially true by using an empty sequence of firings.

Let us show that if $\bar{\sigma}$ is not the zero vector, then there exists at least one vertex $v_i$, such that $\sigma_i > 0$, which is immediately fireable. Consider any $v_i$ such that $\sigma_i > 0$. If it is immediately fireable, the claim is proven. If not, construct, as before, the token-free backwards subgraph. On its boundary we find at least one fireable vertex, say $v_j$. There is a token-free path $v_j \to v_k \to \cdots \to v_s \to v_i$ leading from $v_j$ to $v_i$, and since each arc along the path has no tokens on it, the inequality chain

$$\sigma_j \geqslant \sigma_k \geqslant \cdots \geqslant \sigma_s \geqslant \sigma_i > 0$$

holds, proving $v_j$ to be immediately fireable vertex with $\sigma_j > 0$. Let us fire $v_j$ now.

Consider the resulting marking $M^{(1)}$ generated by this firing and the new firing numbers

$$\sigma_k^{(1)} = \begin{cases} \sigma_k & \text{if } \ k \neq j \\ \sigma_k - 1 & \text{if } \ k = j. \end{cases}$$

It is obvious that

$$M^{(1)}(e) + \sigma_i^{(1)} - \sigma_j^{(1)} = M(e) + \sigma_i - \sigma_j$$

for all $v_i \xrightarrow{e} v_j$, and therefore

$$M^{(1)}(e) + \sigma_i^{(1)} - \sigma_j^{(1)} \geqslant 0.$$

We may now look for a $v_l$—such that $\sigma_l^{(1)} > 0$—which is immediately fireable, fire it, generating $M^{(2)}$ and $\sigma_k^{(2)}$ etc., until we reach a final marking $M^{(p)}$ and $\sigma_i^{(p)} = 0$, for all $i = 1, 2,..., n$. Clearly, $M'(e) = M^{(p)}(e)$ and every $v_i$ was fired $\sigma_i$ times. Q.E.D.

THEOREM 6. *If $\tilde{\sigma}$ is a firing sequence, for a graph whose underlying undirected graph is connected, and this sequence leads back to the initial marking $M$, then all vertices have been fired an equal number of times.*

*Proof.* We have $M(e) = M(e) + \sigma_i - \sigma_j$ for every edge $v_i \xrightarrow{e} v_j$. Thus, every two adjacent vertices have been fired the same number of times. Since the underlying undirected graph is connected, all vertices have been fired the same number of times. Q.E.D.

THEOREM 7. *Let $M$ be a live marking. There exists a firing sequence leading from $M$ to itself, in which every vertex fires exactly once.*

*Proof.* Choose $\sigma_i = 1$ for all $i = 1, 2,..., n$. Our theorem is now an immediate corollary of Theorem 5. Q.E.D.

The following two theorems serve as a basis for the two algorithms described in this paper.

Consider the system of inequalities

$$\left.\begin{array}{l} u_i - u_j + N(e) \leqslant 0 \text{ for all } v_i \xrightarrow{e} v_j \\ \text{where the } N(e)\text{'s are given integers.} \end{array}\right\} (A)$$

A solution vector $\bar{u} = (u_1, u_2, ..., u_n)$ is a vector of values which satisfies System $A$.

THEOREM 8. *Either there is a nonnegative integral solution $\bar{u} \geqslant 0$ to $A$ or there exists a directed circuit*

$$C = v_{i_1} \xrightarrow{e_{j_1}} v_{i_2} \xrightarrow{e_{j_2}} \cdots v_{i_p} \xrightarrow{e_{j_p}} v_{i_1}$$

*such that*

$$\sum_{k=1}^{p} N(e_{j_k}) > 0.$$

*Proof.* First, observe that the alternatives do exclude each other, because if we have a solution $\bar{u}$ to $A$ we may sum $A$'s inequalities along any directed circuit

$$C = v_{i_1} \xrightarrow{e_{j_1}} v_{i_2} \xrightarrow{e_{j_2}} \cdots \longrightarrow v_{i_p} \xrightarrow{e_{j_p}} v_{i_1}.$$

Observing that

$$(u_{i_1} - u_{i_2}) + (u_{i_2} - u_{i_3}) + \cdots + (u_{i_p} - u_{i_1}) = 0,$$

we obtain

$$\sum_{k=1}^{p} N(e_{j_k}) \leqslant 0.$$

We attempt to find a solution $\bar{u}$ to $A$ by a labeling procedure.

Each vertex $v_i$ will be labeled by a pair $(p_i, u_i)$, whesre $p_i$ is a vertex number $1 \leqslant p_i \leqslant n$ to be called $i$'s predecessor, or $p_i \leqslant 0$ in which case $i$ has no predecessor. $u_i$ is an integer or the special value $-\infty$, understood to be smaller than any other finite integer. We start by labeling all vertices $(0, -\infty)$. The labeling procedure proceeds as follows:

(a) Pick an $i$ such that $u_i = -\infty$. If none exists, the process terminates. If one such $i$ is found, set $p_i = -1$, $l \leftarrow i$, $u_i \leftarrow 0$. Then $v_l$ will be called the labeling vertex.

(b) Check all edges such that $v_l \xrightarrow{e} v_k$. If there is any such edge which satisfies $u_l - u_k + N(e) > 0$, set $j \leftarrow k$ and proceed to Step (d). Otherwise, continue.

(c) (Retrace) If $p_l = -1$ (first in chain), set $p_l \leftarrow 0$ and go to Step (a). Otherwise, set $i \leftarrow p_l$, $p_l \leftarrow 0$, $l \leftarrow i$ (set $l$'s predecessor to be the labeling vertex), and return to Step (b).

(d) (Label) If $p_j \neq 0$, go to Step (e). Otherwise, set $p_j \leftarrow l$, $u_j \leftarrow u_l + N(e)$, $l \leftarrow j$ and return to Step (b).

(e) A "positive" circuit has been found. It can be reconstructed by tracing back predecessors: $l, p_l, p_{p_l}, \ldots$, etc., until we reach $j$.

First, let us show that the process terminates. Each time a vertex is labeled or relabeled, its value is determined by a simple path from the vertex whose label is $(-1, 0)$ (there is one such vertex at a time). Since there are finitely many such start vertices, and finitely many simple paths from each of them to a given vertex, a vertex can be labeled a finite number of times only. Since each vertex can help avoid termination in Step (a) only once, no loops are possible, and the process must terminate.

Assume that the process terminates at (a). All the $u$'s are finite. Take any inequality $u_i - u_j + N(e) \leqslant 0$ for $v_i \xrightarrow{e} v_j$. Consider the last time vertex $v_i$ was retraced. At this time, it was verified that for all $j$ such that $v_i \xrightarrow{e} v_j$, $u_i - u_j + N(e) \leqslant 0$. Since then, $u_i$ has not changed, while $u_j$ could only increase. Hence, if the algorithm terminates at Step (a), a solution $\bar{u}$ to $A$ has been found. To make it nonnegative one may subtract from $u_i$ the minimal one.

Suppose, alternatively, that the process terminates at Step (e). Then, clearly, we have a directed circuit

$$C = v_{i_1} \xrightarrow{e_{j_1}} v_{i_2} \xrightarrow{e_{j_2}} \cdots v_{i_p} \xrightarrow{e_{j_p}} v_{i_1},$$

such that $v_{i_1}$ has labeled $v_{i_2}$, etc., and in the end $v_{i_p}$ attempts to label $v_{i_1}$ again. Thus,

$$\begin{aligned} u_{i_2} &= u_{i_1} + N(e_{j_1}), \\ u_{i_3} &= u_{i_2} + N(e_{j_2}) \\ &\vdots \\ u_{i_p} &= u_{i_{p-1}} + N(e_{j_{p-1}}), \\ u_{i_1} &< u_{i_p} + N(e_{j_p}). \end{aligned}$$

By summing up these relations, we obtain:

$$\sum_{k=1}^{p} N(e_{j_k}) > 0. \qquad \text{Q.E.D.}$$

The following theorem is a corollary of Theorem 8:

Consider the system of inequalities

$$\left.\begin{array}{l} u_i - u_j + N(e) \geqslant 0 \text{ for all } v_i \xrightarrow{e} v_j \\ \text{where the } N(e)\text{'s are given integers.} \end{array}\right\} (B)$$

THEOREM 9. *Either there is a nonnegative integral solution $\bar{u} \geqslant 0$ to $B$ or there exists a directed circuit*

$$C = v_{i_1} \xrightarrow{e_{j_1}} v_{i_2} \xrightarrow{e_{j_2}} \cdots v_{i_p} \xrightarrow{e_{j_p}} v_{i_1}$$

*such that*

$$\sum_{k=1}^{p} N(e_{j_k}) < 0.$$

Given two markings $M$ and $M'$, we say that $M'$ *lies above* $M$ if for all edges $v_i \xrightarrow{e} v_j$, $M(e) \leqslant M'(e)$. We say that $M'$ has *circuit count not lower* than $M$ if for every directed circuit $C$ $\langle M' \mid C \rangle \geqslant \langle M \mid C \rangle$[2]. Our next algorithm will test whether a given marking $M'$ can be brought by a legal sequence of firings to lie above a given marking $M$, and in case such a sequence of firing exists, will produce such a sequence. It is clear that a necessary condition for $M'$ is that its circuit count is not lower than $M$. The algorithm will be described in the proof of the sufficiency of this condition.

THEOREM 10. *A live marking $M'$ can be brought to lie above $M$ if and only if its circuit count is not lower than that of $M$.*

*Proof.* As noted, it is only necessary to prove the 'if' part.

For every edge $e$ set $N(e) = M'(e) - M(e)$. For these values assigned to $N(e)$, attempt to solve System B of inequalities. By Theorem 9 (through the algorithm as in the proof of Theorem 8) we either have a solution $\bar{u} \geqslant 0$, or there exists a circuit

$$C = v_{i_1} \xrightarrow{e_{j_1}} v_{i_2} \xrightarrow{e_{j_2}} \cdots v_{i_p} \xrightarrow{e_{j_p}} v_{i_1}$$

for which $\sum_{k=1}^{p} N(e_{j_k}) < 0$. In the latter case, we have

$$\sum_{k=1}^{p} M'(e_{j_k}) < \sum_{k=1}^{p} M(e_{j_k})$$

which means that $\langle M' \mid C \rangle < \langle M \mid C \rangle$, a contradiction. Thus, the former must hold—namely, for every edge $v_i \xrightarrow{e} v_j$,

$$u_i - u_j + M'(e) - M(e) \geqslant 0,$$

or

$$u_i - u_j + M'(e) \geqslant M(e) \geqslant 0.$$

By Theorem 5 (and through the algorithm described in its proof), there exists a legal sequence of firings which leads from $M'$ to $M''$ defined by

$$M''(e) = u_i - u_j + M'(e),$$

and $M''$ lies above $M$. Q.E.D.

In fact, the proof of Theorem 10 is of more value than the theorem itself. For if we are given a live $M'$ and any $M$, the proof provides a procedure for testing if $M'$ can be brought to lie above $M$ without checking all circuits for their circuit count.

THEOREM 11. *In a strongly connected graph, $M$ can be derived by firing from a live $M'$ if and only if they have identical circuit counts, i.e., for every $C$ $\langle M' \mid C \rangle = \langle M \mid C \rangle$.*

[2] As defined earlier, $\langle M \mid C \rangle$ is the token count placed by the marking $M$ on the circuit $C$.

*Proof.* Again the "only if" is immediate. If $M'$ satisfies the condition, then its circuit count is not lower than that of $M$, and by Theorem 10, it leads to a marking $M''$ which lies above $M$. If $M'' \neq M$, then there exists an edge $e$ such that $M''(e) > M(e)$, but this edge can be completed to a circuit in which the circuit count equality between $M''$ and $M$ (and, therefore, between $M'$ and $M$) is violated. Q.E.D.

THEOREM 12. *In a strongly connected graph if a live marking $M'$ can produce $M$ (through a legal sequence of firings), then $M$ can produce $M'$.*

*Proof.* If $M'$ can produce $M$, then by Theorem 11 they have identical circuit counts. Thus, $M$ is live, and again by Theorem 11 $M$ can produce $M'$. Q.E.D.

This shows that the live markings of a strongly connected graph partition into equivalence classes. We have shown an algorithm for testing whether two markings belong to the same equivalence class. Let us refer to each equivalence class as a *family*.

## IV. THE MAXIMUM MARKING

Simple examples show that firing may change the overall number of tokens in the graph considerably. If we interpret marked graphs as plans of continuous production which allows concurrent processing, then the number of tokens bears direct relation to the number of resources needed at a particular instance. The maximum marking belonging to a given family gives a bound on the maximum resource requirement of a particular organization of a process.

We solve this programming by making use of its dual which is a flow problem.

Let us consider a flow in the underlying strongly connected graph which is a circulation flow, i.e., with no sources or sinks. We impose a lower bound of one unit on each edge, so that the flow is given by integers (the computation ensures that by starting with an integral flow and making integral changes only). We denote the flow in $v_i \xrightarrow{e} v_j$ by $\phi(e)$. The flow must satisfy the conservation law at the vertices, i.e., the total flow entering a vertex is equal to the total flow emanating from it.

The total cost $T_M(\phi)$ of the flow is computed by using any particular marking $M$ of the family as a set of prices on the edges; that is,

$$T_M(\phi) = \sum_{e \in E} M(e) \cdot \phi(e).$$

Note that if $M$ and $M'$ belong to the same family, then $T_M(\phi) = T_{M'}(\phi)$. This follows from the fact that vertex firing does not change the total cost, since the reduction of the cost on the incoming edges is compensated by the increase of the cost on the outgoing edges. Thus, the minimum flow solution is independent of the initial marking of the family which is chosen for measuring the flow.

We solve the minimum flow problem in two phases.

*Phase I.* Establish a feasible flow. This is done by initially setting $\phi(e) = 0$ for all edges $e$. Now repeatedly apply the following step:

If there is no edge $e$ for which $\phi(e) = 0$, stop. If such an edge is found, locate a directed circuit which passes through $e$ and increase the flow on all its edges by 1.

*Phase II.* Define a new graph $\tilde{G}(V, \tilde{E})$ as follows: $\tilde{E}$ contains all edges $e \in E$ for which $\phi(e) > 1$; in addition, for every edge $v_i \xrightarrow{e} v_j$ in $G$, $\tilde{G}$ contains a counterpart $v_i \xleftarrow{e'} v_j$ (a new edge). Now define a function $N(e)$ on the edges of $\tilde{G}$ as follows:

$$N(e) = \begin{cases} M(e) \text{ if } e \in E, \\ -M(e') \text{ if } e \text{ is the counterpart of } e'(e' \in E). \end{cases}$$

Apply the algorithmic part of the proof of Theorem 8 to System $A$ defined by $N(e)$ for $\tilde{G}$. If we fail to find a solution to System $A$, then we have located a directed circuit $C'$ in $\tilde{G}$

$$C' = v_{i_1} \xrightarrow{e_{j_1}} v_{i_2} \xrightarrow{e_{j_2}} \cdots v_{i_p} \xrightarrow{e_{j_p}} v_{i_1}$$

for which

$$\sum_{k=1}^{p} N(e_{j_k}) > 0.$$

The directed circuit $C'$ (in $\tilde{G}$) corresponds to a circuit $C$ in $G$. $C$ may not be directed, since some of the edges of $C'$ may be counterparts of edges in $G$ and, therefore, are oppositely directed. Thus,

$$\sum_{k=1}^{p} N(e_{j_k}) = \sum_{e \in R} M(e) - \sum_{e \in W} M(e),$$

where $R$ is the set of edges in $C$ whose direction is as in $C'$, and $W$ is the set of edges in $C$ whose direction is opposite to that in $C'$. Thus,

$$\sum_{e \in R} M(e) > \sum_{e \in W} M(e).$$

Now, by reducing the flow in all edges of $R$ by one unit and increasing it in all edges of $W$ by one unit, the conservation law is kept in all the vertices, and the total cost reduces. We redefine $\tilde{G}$ and repeat. Since the total cost is bounded from below by $\sum_{e \in E} M(e)$, sooner or later a solution $\bar{u} \geqslant 0$ to the current System $A$ must be found. Thus, we have,

$$u_i - u_j + N(e) \leqslant 0 \text{ for all } v_i \xrightarrow{e} v_j \text{ in } \tilde{G}.$$

For every edge $v_i \xrightarrow{e} v_j$ in $G$ we have $v_i \xleftarrow{e'} v_j$ in $\tilde{G}$. Therefore, $u_j - u_i - M(e) \leqslant 0$ or $M(e) + u_i - u_j \geqslant 0$. By Theorem 5 (and through the algorithm in its proof) the marking $M'(e) = M(e) + u_i - u_j$ is in the same family. Furthermore, if $\phi(e) > 1$ then $v_i \xrightarrow{e} v_j$ belongs to $\tilde{G}$. Thus, $u_i - u_j + N(e) \leqslant 0$, which implies

$$u_i - u_j + M(e) \leqslant 0,$$

viz.,

$$M'(e) = 0.$$

Thus, the flow is measured only on edges $e$ for which $\phi(e) = 1$. This proves that the flow is minimum, for it has reached the lower bound on all the edges in which it is measured, and that the marking $M'$ is maximum, as the sum $\sum_{e \in E} M(e)$, for any marking $M$ in the family, is a lower bound on the cost $\sum_{e \in E} M(e)\,\phi(e)$, and here $\sum_{e \in E} M'(e)$ is the value of the cost.

Note that the algorithm described here is a minimum cost circulation algorithm. Any other minimum cost circulation algorithm, such as that of Ford and Fulkerson [4] could serve as well. However, we believe that our approach is the natural one in this framework. The problem of finding a minimum marking (which may be a convenient initial state for the system) can be solved in a similar way together with the dual maximum flow problem.

So far we were not able to solve the problem of finding a maximum live and safe marking for a given strongly connected graph, if no restriction to a given family is imposed.

## References

1. C. A. Petri, "Kommunikation mit Automaten," Schriften des Rheinisch-Westfalischen Institutes für Instrumentelle Mathematik an der Universität Bonn, Hft. 2, Bonn, 1962. Translated into English by Project ISTP: "Communication with Automata," translated by Clifford F. Greene, Jr., a Supplement to Technical Documentary Report No. 1 prepared for Rome Air Development Center #AF 30 (602)-3344, 1965.
2. A. W. Holt, *et al.*, "Information System Theory Project," Technical Report for Rome Air Development Center No. RADC-TR-68-305, prepared by Applied Data Research, Inc., Princeton, N. J., 1968. Available from the Department of Commerce Clearing House, Springfield, Virginia, as report No. AD 676-972.
3. R. M. Shapiro and H. Saint, A new approach to optimization of sequencing decisions, Annual Review in Automatic Programming, Pergamon Press, Headington Hill Hall, Oxford, OX3 OBW, England, Vol. 6, Part 5, 1970.
4. L. R. Ford, Jr. and D. R. Fulkerson, "Flows in Networks," Princeton University Press, Princeton, N. J., 1962.
5. A. W. Holt and F. Commoner, Marked graphs and linear programming, unpublished notes.

6. A. W. HOLT AND F. COMMONER, "Events and Conditions," Information systems theory project. Research Report of Applied Data Research, Lakeside Office Park, Wakefield, Massachusetts 01880, 1970.
7. H. J. GENRICH, "Einfache Nicht-Sequentielle Prozesse," Gesellschaft für Mathematik und Datenverarbeitung, Birlinghoven, West Germany, 1970.
8. R. M. KARP AND R. E. MILLER, "Properties of a Model for Parallel Computations: Determinacy, Terminations, Queueing," *SIAM J. Appl. Math.* **14** (1966), 1300–1411.
9. R. REITER, "Scheduling Parallel Computations," *JACM* **14** (1968), 590–599.